

UNCLASSIFIED

AD NUMBER

AD835359

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; MAR 1968. Other requests shall be referred to Air Force Technical Application Center, VELA Seismological Center, Washington, DC 20333. This document contains export-controlled technical data.

AUTHORITY

usaf ltr, 25 jan 1972

THIS PAGE IS UNCLASSIFIED



AD835359

ITERATIVE TECHNIQUES  
FOR THE SOLUTION OF  
FREQUENCY-DOMAIN FILTER SETS  
LARGE-ARRAY SIGNAL AND NOISE ANALYSIS  
Special Scientific Report No. 16

Prepared by  
Peter R. Fenner.

Frank H. Binder, Program Manager  
1-214-238-3491

TEXAS INSTRUMENTS INCORPORATED  
Science Services Division  
P.O. Box 5621  
Dallas, Texas 75222

JUL 15 1968

Contract No. AF 33(657)-16678

This document is subject to the provisions of the Arms Control and each  
transmittal to foreign governments or international organizations may be  
made only with prior approval of the Department of Defense.

Prepared for

AIR FORCE TECHNICAL APPLICATIONS CENTER  
Washington, D.C. 20333

*attn: Vela Semiological Center*  
Sponsored by

ADVANCED RESEARCH PROJECTS AGENCY  
ARPA Order No. 599  
AFTAC Project No. VT/6707

22 March 1968

science services division

61

**BEST  
AVAILABLE COPY**



ITERATIVE TECHNIQUES  
FOR THE SOLUTION OF  
FREQUENCY-DOMAIN FILTER SETS  
LARGE-ARRAY SIGNAL AND NOISE ANALYSIS  
Special Scientific Report No. 16

Prepared by  
Peter R. Fenner

Frank H. Binder, Program Manager  
1-214-238-3491

TEXAS INSTRUMENTS INCORPORATED  
Science Services Division  
P. O. Box 5621  
Dallas, Texas 75222

Contract No. AF 33(657)-16678

Prepared for  
AIR FORCE TECHNICAL APPLICATIONS CENTER  
Washington, D. C. 20333

Sponsored by  
ADVANCED RESEARCH PROJECTS AGENCY  
ARPA Order No. 599  
AFTAC Project No. VT/6707

22 March 1968



## TABLE OF CONTENTS

Section	Title	Page
I	INTRODUCTION AND SUMMARY	I-1
II	MULTICHANNEL FREQUENCY FILTER DESIGN	II-1
III	SUMMARY OF GRADIENT METHODS	III-1
	A. STEEPEST-DESCENT METHOD	III-1
	B. FINITE PROCEDURE	III-2
IV	ADAPTIVE UPDATE METHODS	IV-1
	A. EXACT INVERSE MATRIX UPDATE METHOD	IV-2
	B. STEEPEST-DESCENT METHOD	IV-2
	C. COMMENTS	IV-5
V	REQUIRED COMPUTATIONS	V-1
	A. GAUSSIAN ELIMINATION METHOD	V-1
	B. STEEPEST-DESCENT METHOD	V-2
	C. FINITE PROCEDURE	V-3
	D. EXACT INVERSE METHOD FOR SINGLE-CHANNEL PREDICTION	V-3
	E. COMPARISON	V-3
VI	CONCLUSIONS	VI-1

## LIST OF APPENDIXES

Appendix	Title
A	GRADIENT METHODS
B	EXACT INVERSE EQUATION
C	NUMERICAL RESULTS



## SECTION I

### INTRODUCTION AND SUMMARY

Computation of high-resolution wavenumber spectra includes, as an intermediate step, the solution of a set of Hermitian equations. This set of equations has the form of a least-squares multichannel frequency-domain filter design equation which can be expressed in vector-matrix notation as

$$H \bar{a} = \bar{b} \quad (1-1)$$

where

- $H$  =  $n \times n$  nonsingular Hermitian power spectral matrix
- $\bar{b}$  =  $n \times 1$  known complex column vector of output power spectra
- $\bar{a}$  =  $n \times 1$  unknown complex column vector of filter weights

This report investigates three techniques of solving for the unknown vector  $\bar{a}$ : the method of conjugate gradients, steepest-descent method, and exact-inverse method. The object is to determine the accuracy and computational complexity of each technique.

Previously, problems of this type were solved using a direct numerical matrix technique such as the Gaussian elimination method\* or the square-root method.\*\*

---

\* Fox, L., 1965: An Introduction to Numerical Linear Algebra, Oxford University Press, New York, p. 205-213.

\*\* Faddeeva, V.N., 1959: Computational Methods of Linear Algebra, Dover Publications, Inc., New York.



To obtain a solution, the Gaussian elimination method requires only that the matrix be nonsingular, while the square-roots method requires the matrix to be nonsingular and Hermitian. It was apparent that neither a knowledge of the form of the H matrix nor an estimate of the solution vector  $\bar{a}$  would significantly simplify the computation involved with either of these techniques. Since both the general form of the H matrix and an estimate of the solution vector were available, an investigation of two iterative techniques (steepest-descent and conjugate-gradients) which appeared to benefit computationally from this knowledge was initiated. Found during this investigation was a theorem from linear algebra which analytically expresses the inverse of the particular H matrix used to generate high-resolution f-k filter sets. Application of this analytical inverse greatly reduced the computation needed to generate high-resolution f-k filter sets.

The high-resolution wavenumber filter sets are designed using a power spectral matrix H of the form

$$H = \left[ \xi I + \bar{x} \bar{x}^* \right] \quad (1-2)$$

where

$I$  =  $n \times n$  identity matrix

$\bar{x}$  =  $n \times 1$  column vector of the channel transforms

$\bar{x}^*$  =  $n \times 1$  row vector of the conjugate channel transforms

Using the exact inverse equation, the filter vector  $\bar{a}$  can be computed easily as

$$\bar{a} = \frac{1}{\xi} \left[ \bar{b} - \frac{(\bar{x}^* \bar{b})}{(\xi + \bar{x}^* \bar{x})} \bar{x} \right] \quad (1-3)$$



This computation requires much fewer calculations than a single cycle of the iterative techniques studied and is orders of magnitude simpler than the Gaussian elimination technique.

Further application of the exact inverse equation arises when a power spectral matrix is generated using an exponentially weighted series of transform vectors:

$$H = \sum_{j=1}^N \alpha^{N-j} \bar{x}_j \bar{x}_j^* \quad 0 < \alpha \leq 1 \quad (1-4)$$

The inverse of  $H$  can be generated iteratively by writing

$$H_{j+1} = \alpha H_j + \bar{x}_{j+1} \bar{x}_{j+1}^* \quad (1-5)$$

The exact inverse equation is

$$H_{j+1}^{-1} = \frac{1}{\alpha} \left[ H_j^{-1} - \left( \frac{1}{\alpha + \bar{x}_{j+1}^* H_j^{-1} \bar{x}_{j+1}} \right) H_j^{-1} \bar{x}_{j+1} \bar{x}_{j+1}^* H_j^{-1} \right] \quad (1-6)$$

This formulation for the inverse of a power spectral matrix is especially useful when the transform vectors  $\bar{x}_j$  are available sequentially as they would be in a practical situation. The inverse of the power spectral matrix can be updated as each transform vector becomes available; and from this inverse matrix, many frequency-domain filter sets can be designed at each iteration. The ability to track a spectral matrix adaptively facilitates the coherent frequency-domain processing of small arrays with nonstationary noise fields.



**BLANK PAGE**



## SECTION II

### MULTICHANNEL FREQUENCY FILTER DESIGN

This report investigates three iterative techniques which offer computational savings in the design of multichannel frequency-domain filter sets. This study was initiated to reduce the computational complexity involved in generating high-resolution  $f\text{-}\vec{k}$  spectra, thereby enhancing the technique's capability of a real-time detection and location. The following discussion briefly considers the general multichannel frequency-domain design problem then proceeds to the special problem of high-resolution  $f\text{-}\vec{k}$  filter design.

The general multichannel filter-design equation can be written in vector-matrix notation as

$$\begin{bmatrix} \overline{x_1^* x_1} & \cdot & \cdot & \cdot & \overline{x_1^* x_n} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \overline{x_n^* x_1} & \cdot & \cdot & \cdot & \overline{x_n^* x_n} \end{bmatrix} \begin{bmatrix} a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{bmatrix} = \begin{bmatrix} \overline{x_1^* S} \\ \cdot \\ \cdot \\ \cdot \\ \overline{x_n^* S} \end{bmatrix} = \bar{b} \quad (2-1)$$

where  $\overline{x_i^* x_j}$  is the estimated crosspower (for  $i \neq j$ ) or the estimated autopower (for  $i = j$ ) between the  $i^{\text{th}}$  and the  $j^{\text{th}}$  channels and where  $\overline{x_j^* S}$  is the cross-power estimate between the desired signal output and the  $j^{\text{th}}$  data channel. These autopower and crosspower spectra usually are estimated from a sample of multichannel data using one of two general approaches.



One method is generating the autocorrelations and crosscorrelations from the time-domain data segment and transforming these correlations to obtain the desired power spectra. The other method involves transforming segments of the data sample and stacking and/or smoothing these transforms to obtain the desired power spectra. The second method is pertinent to this report as it is computationally more efficient than the first method.

The transform operation itself is not discussed, but the data vector

$$\bar{x}_l = \begin{bmatrix} x_{1l} \\ \cdot \\ \cdot \\ \cdot \\ x_{nl} \end{bmatrix} \quad (2-2)$$

is assumed to be the colateral discrete transforms of all the data channels of the  $l^{\text{th}}$  data segment at a particular frequency. The power spectral matrix at this frequency then can be estimated as the weighted sum of a priori signal power spectral information  $S$  and the transform crossproducts  $\bar{x} \bar{x}^*$  so that

$$\begin{bmatrix} \overline{x_1^* x_1} & \cdot & \cdot & \cdot & \overline{x_1^* x_n} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \overline{x_n^* x_1} & \cdot & \cdot & \cdot & \overline{x_n^* x_n} \end{bmatrix} = H = \xi S + \sum_{l=1}^m \alpha_l \bar{x}_l \bar{x}_l^* \quad (2-3)$$



Using this estimate for the power spectral matrix, the filter design equation can be written as

$$\left[ \xi S + \sum_{l=1}^m \alpha_l \bar{x}_l \bar{x}_l^* \right] \bar{a} = \bar{b} \quad (2-4)$$

Most iterative solution techniques compute a residual vector  $\bar{r}_1$  from an initial guess  $\bar{v}_0$  of the solution vector  $\bar{a}$ :

$$\bar{r}_1 = \bar{b} - H \bar{v}_0 \quad (2-5)$$

In general, the computation of this residual vector requires  $n^2$  complex multiply and add operations, not including the computations needed to form the matrix  $H$ . In the special case, using Equation 2-3 to form the matrix  $H$ , the residual vector can be computed directly without explicitly forming the  $H$  matrix. This is done by substituting Equation 2-3 into Equation 2-5 and using a different order of computation, as shown in the following equation.

$$\begin{aligned} \bar{r} &= \left[ \xi S + \sum_{l=1}^m \alpha_l \bar{x}_l \bar{x}_l^* \right] \bar{v}_0 - \bar{b} \\ &= \xi S \bar{v}_0 + \sum_{l=1}^m \alpha_l x_l \left( \bar{x}_l^* \bar{v}_0 \right) - \bar{b} \end{aligned} \quad (2-6)$$

To form the  $H$  matrix using Equation 2-3,  $mn^2$  complex multiply and add operations are required. Equation 2-6 can compute the residual using  $n^2 + 3mn$  complex multiply and add operations by first forming the vector dot product  $\bar{x}_l^* \bar{v}_0$ , then performing the scalar multiplication, and summing over the resulting vectors. This compares to  $(m+1)n^2$  complex multiply and add operations needed to form  $H$  separately and then compute  $\bar{r}$ .



Further computational savings can be obtained if the signal-model matrix  $S$ , which contributes the  $n^2$  term, can be expressed as a diagonal matrix (uncorrelated noise)

$$S = \begin{bmatrix} s_1 & 0 & . & . & . & 0 \\ 0 & s_2 & & & & \\ . & & . & & & \\ . & & & . & & \\ . & & & & . & \\ 0 & & & & & s_n \end{bmatrix} \quad (2-7)$$

or the sum of vector products (several plane waves)

$$S = \sum_{k=1}^p c_k \bar{z}_k \bar{z}_k^* \quad (2-8)$$

For diagonal matrix  $S$ ,  $(3m + 1)n$  complex multiply and add operations are required to compute  $\bar{r}$ . If  $S$  were defined by Equation 2-8,  $3(m + p)n$  multiply and add operations would be needed. Since using the Gaussian elimination method to solve Equation 2-1 requires at least  $2n^3$  operations, significant savings in computation will result from using iterative solution techniques, provided only a few iterations are required.

The special problem of designing high-resolution  $f\text{-}\bar{k}$  filter sets uses a simple form of the power spectral matrix where

$$H = [\xi I + \bar{x} \bar{x}^*] \quad (2-9)$$



This form results from the transient nature of the detected signals, but these likely will not be present in several different transform segments of the data sample. The signal model is assumed to be equalized and uncorrelated from channel to channel; therefore,  $S = I$  (the identity matrix), and

$$\bar{b} = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad (2-10)$$

A theorem from linear algebra, which will be called the exact inverse equation (Appendix B), reduces the computation of high-resolution f-k filter sets to approximately  $3n$  complex multiply and add operations (where  $n$  is the number of channels). The exact inverse equation can be expressed generally as

$$\left[ A + \bar{u} \bar{v}^* \right]^{-1} = A^{-1} - \frac{A^{-1} \bar{u} \bar{v}^* A^{-1}}{1 + \bar{v}^* A^{-1} \bar{u}} \quad (2-11)$$

where

$A = n \times n$  nonsingular complex matrix

$\bar{u} = n \times 1$  column vector

$\bar{v} = n \times 1$  column vector

$\bar{v}^* =$  conjugate transpose of  $\bar{v}$

$\left[ A + \bar{u} \bar{v}^* \right]$  is nonsingular



Applying this theorem to the high-resolution  $f\text{-}\vec{k}$  filter design yields

$$\vec{a} = \frac{1}{\xi} \left[ I - \frac{\vec{x} \vec{x}^*}{\xi + \vec{x}^* \vec{x}} \right] \vec{b} = \frac{1}{\xi} \left[ \vec{b} - \left( \frac{\vec{x}^* \vec{b}}{\xi + \vec{x}^* \vec{x}} \right) \vec{x} \right] \quad (2-12)$$

Pointing out the simplicity of the calculations, this vector-matrix equation can be rewritten as

$$\begin{aligned} a_1 &= \frac{1}{\xi} \left[ 1 - \frac{x_1^* x_1}{\xi + \vec{x}^* \vec{x}} \right] \\ a_j &= \frac{-1}{\xi} \left[ \frac{x_1^* x_j}{\xi + \vec{x}^* \vec{x}} \right] \quad j = 2, 3, \dots, n \end{aligned} \quad (2-13)$$

The vector product  $\vec{x}^* \vec{x}$  is real and needs to be computed only once for each transform segment. Remaining calculations can be accomplished with the equivalent of  $2n$  more complex multiply and adds (CMPA's). The required computations are discussed in Section V. The total computation necessary for the exact inverse method is roughly  $3n$  complex multiply and add operations compared with approximately  $12n$  CMPA's for one iteration of the steepest-descent method and  $24n$  CMPA's for one iteration of the finite procedure. Thus, the exact inverse technique is the most straightforward and efficient procedure to design high-resolution  $f\text{-}\vec{k}$  filter sets. It is possible to manipulate the exact inverse equation to update the inverse of a power spectral matrix; this area is covered in the section on adaptive update methods.



### SECTION III

#### SUMMARY OF GRADIENT METHODS

Based on the gradient of a particular quadratic form, two iterative design procedures — steepest-descent and conjugate-gradient methods — were investigated during this study. While these two approaches are not the most efficient means of obtaining high-resolution  $f\text{-}k$  filter sets, they are applicable to the more general filter-design problems. The details of these methods are described in Appendix A.

##### A. STEEPEST-DESCENT METHOD

The numerical algorithm generated using the steepest-descent approach is

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{r}_k \quad (3-1)$$

$$\bar{r}_k = \bar{b} - H \bar{v}_k \quad (3-2)$$

$$p_k = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_k^* H \bar{r}_k} \quad (3-3)$$

A simpler procedure results when the form of the  $H$  matrix is

$$H = \xi I + \sum_{\ell=1}^m \omega_{\ell} \bar{x}_{\ell} \bar{x}_{\ell}^* \quad (3-4)$$

The matrix operations become vector operations, and the algorithm is

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{r}_k \quad (3-1)$$

$$\bar{r}_k = \bar{b} - \xi \bar{v}_k - \sum_{\ell=1}^m \omega_{\ell} \bar{x}_{\ell} (\bar{x}_{\ell}^* \bar{v}_k) \quad (3-5)$$





$$p_k = \frac{\bar{r}_k^* \bar{r}_k}{\xi \bar{r}_k^* \bar{r}_k + \sum_{l=1}^m \omega_l \bar{z}_l^* \bar{z}_l} \quad (3-6)$$

$$\bar{z}_{lk} = \bar{x}_l^* \bar{r}_k \quad (3-7)$$

While the notation of Equations 3-5, 3-6, and 3-7 is more complicated than that of Equations 3-2 and 3-3, the amount of computation needed has been reduced. The reduction in computation is obtained by performing the vector-dot-product operations  $\bar{x}_l^* \bar{v}_k$  and  $\bar{x}_l^* \bar{r}_k$  first, leaving only scalar operations to be performed. An analysis of the amount of computation required is presented in a later section of this report.

#### B. FINITE PROCEDURE (CONJUGATE-GRADIENT METHOD)

The computation cycle evolving this procedure is

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{w}_k \quad (3-1)$$

$$\bar{r}_k = \bar{b} - H \bar{v}_k \quad (3-2)$$

$$p_k = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_k^* H \bar{r}_k} \quad (3-3)$$

$$q_{k-1} = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_{k-1}^* \bar{r}_{k-1}} \quad (3-8)$$

$$\bar{w}_k = \bar{r}_k + q_{k-1} \bar{w}_{k-1} \quad (3-9)$$

The first cycle is computed using  $\bar{w}_0 = \bar{r}_0$ .



Again, consider the special case where

$$H = \xi I + \sum_{\ell=1}^m w_{\ell} \bar{x}_{\ell} \bar{x}_{\ell}^* \quad (3-4)$$

The computational cycle becomes

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{w}_k \quad (3-1)$$

$$\bar{r}_k = \bar{b} - \xi \bar{v}_k - \sum_{\ell=1}^m w_{\ell} \bar{x}_{\ell} (\bar{x}_{\ell}^* \bar{v}_k) \quad (3-5)$$

$$u_k = \bar{r}_k^* \bar{r}_k \quad (3-10)$$

$$z_{\ell k} = \bar{x}_{\ell}^* \bar{r}_k \quad (3-7)$$

$$p_k = u_k / \left( \xi u_k + \sum_{\ell=1}^m w_{\ell} z_{\ell k}^* z_{\ell k} \right) \quad (3-11)$$

$$q_{k-1} = u_k / u_{k-1} \quad (3-12)$$

$$\bar{w}_k = \bar{r}_k + q_{k-1} \bar{w}_{k-1} \quad (3-9)$$

For the first iteration,  $\bar{w}_0 = \bar{r}_0$  and  $q_{k-1}$  is not computed.

Notation again becomes more cumbersome, but the computation needed is much less than that required to form the autopower-crosspower matrix and then solve for the desired filter set. Computational savings are obtained by carrying out the same computations in a different order. A detailed analysis of the numerical operations involved is presented in a later section of this report.



## SECTION IV

### ADAPTIVE UPDATE METHODS

In many practical applications, all individual vectors  $x_j$  which make up the estimated covariance matrix  $H$  are not simultaneously available. The various samples must be stored before they can be used in the filter design. One method stores all the needed vectors in a computer memory and uses one previously mentioned algorithm to compute a new filter set each time a new vector is received. Another method iteratively updates either the filter set or the covariance-matrix inverse each time a new vector is received. Iterative updating reduces the storage and computation required but is restricted to using a class of memory functions. This class includes the exponential-weighting function which is discussed in this section.

Two of the techniques previously discussed can be formulated as iterative-updating techniques which weight previous transform vectors with an exponentially decreasing function. Exponential weighting can be obtained by computing a weighted sum of the previous covariance estimate and the current transform-vector product; i. e.,

$$H_n = \left[ \alpha H_{n-1} + \bar{x}_n \bar{x}_n^* \right] \quad 0 \leq \alpha \leq 1 \quad (4-1)$$

The  $n + 1$  estimate will be

$$\begin{aligned} H_{n+1} &= \alpha H_n + \bar{x}_{n+1} \bar{x}_{n+1}^* \\ &= \alpha \left[ \alpha H_{n-1} + \bar{x}_n \bar{x}_n^* \right] + \bar{x}_{n+1} \bar{x}_{n+1}^* \\ &= \alpha^2 H_{n-1} + \alpha \bar{x}_n \bar{x}_n^* + \bar{x}_{n+1} \bar{x}_{n+1}^* \end{aligned} \quad (4-2)$$



etc. Thus, a vector  $n$  samples in the past is weighted by  $\alpha^n$ . If  $0 < \alpha < 1$ , the current estimate of the autopower-crosspower matrix weights the current transform vector with weights equal to unity and previous transform vectors with weights of less than unity. These weights (Equation 4-2) become exponentially smaller for the more distant samples. The past transform vectors can be weighted in the desired fashion by the choice of  $\alpha$ , where  $\alpha = \exp(-\beta)$  for  $\beta > 0$ .

#### A. EXACT INVERSE MATRIX UPDATE METHOD

One updating method applies the exact inverse equation directly to Equation 4-1. The inverse of the updated covariance matrix is then

$$H_n^{-1} = \frac{1}{\alpha} \left[ H_{n-1}^{-1} - \frac{H_{n-1}^{-1} \bar{x}_n \bar{x}_n^* H_{n-1}^{-1}}{\alpha + \bar{x}_n^* H_{n-1}^{-1} \bar{x}_n} \right] \quad (4-3)$$

The desired filter weights can be obtained from the matrix multiplication

$$\bar{f} = H_n^{-1} \bar{b} \quad (4-4)$$

In this case, the previously computed covariance matrix is considered to be the noise model. This method requires computing and storage of an entire matrix at each iteration as well as a matrix-vector multiplication for each filter set needed. When only a few filter sets are needed, the steepest-descent technique is computationally more efficient.

#### B. STEEPEST-DESCENT METHOD

A particularly simple scheme is afforded by extending the steepest-descent method. The new filter set is formed from the previous filter set by computing



$$\bar{f}_{n+1} = \bar{f}_n + p_{n+1} \bar{r}_{n+1} \quad (4-5)$$

where the residual vector  $\bar{r}_{n+1}$  is

$$\bar{r}_{n+1} = \bar{b} - H_{n+1} \bar{f}_n \quad (4-6)$$

The matrix

$$H_{n+1} = \alpha H_n + \bar{x}_{n+1} \bar{x}_{n+1}^* \quad (4-7)$$

This allows the residual vector to be expressed as

$$\bar{r}_{n+1} = \bar{b} - \left[ \alpha H_n + \bar{x}_{n+1} \bar{x}_{n+1}^* \right] \bar{f}_n \quad (4-8)$$

$$= (1 - \alpha) \bar{b} + \alpha \left[ \bar{b} - H_n \bar{f}_n \right] - \bar{x}_{n+1} \bar{x}_{n+1}^* \bar{f}_n$$

The quantity  $\bar{b} - H_n \bar{f}_n$  is the residual vector  $\bar{r}_n$  from the previous iteration. The new residual vector is then

$$\bar{r}_{n+1} = (1 - \alpha) \bar{b} + \alpha \bar{r}_n - \bar{x}_{n+1} \bar{x}_{n+1}^* \bar{f}_n \quad (4-9)$$

This new residual vector, used to calculate the new filter weights, is stored to be used in the next residual calculation.



The problem of computing the iteration factor  $p_{n+1}$  is encountered now. Since the matrix  $H$  exists implicitly at each iteration but is not directly available for computation, the iteration factor cannot be computed directly from

$$p_{n+1} = \frac{\bar{r}_{nH}^* \bar{r}_{n+1}}{\bar{r}_{n+1}^* H_{n+1} \bar{r}_{n+1}}$$

An alternative to computing the iteration factor for each cycle is to choose a single value and use it throughout the process. This single value must be small enough to guarantee the numerical stability of the process. This requirement will be satisfied if

$$0 < p \leq \frac{1}{|\lambda_{\max}|}$$

where  $\lambda_{\max}$  is the largest eigenvalue of the matrix  $H_n$ . Since the matrix  $H_n$  is changing continually as new data samples are added, only the upper bound on the maximum eigenvalue can be expressed. A matrix of the form of Equation 4-1 is essentially a weighted sum of vector products

$$H_n = \sum_{j=0}^{\infty} \alpha^j \bar{x}_{n-j} \bar{x}_{n-j}^* \quad \text{for } 0 < \alpha < 1$$

The largest eigenvalue  $\lambda_{\max}^n$  of  $H_n$  is then

$$\lambda_{\max} \leq \left[ \bar{x}_{\max}^* \bar{x}_{\max} \right]$$



where  $\bar{x}_{\max}$  is the largest data vector encountered. This implies that the choice of  $p$  should be in the range

$$0 < p \leq \left( \frac{1}{\bar{x}_{\max}^* \bar{x}_{\max}} \right)$$

Such a conservative choice of the iteration factor sacrifices rapid convergence for numerical stability. This approach requires a priori knowledge of the maximum magnitude of the transform vector. In a practical situation, overestimating the value of  $\bar{x}_{\max}^* \bar{x}_{\max}$  would be necessary to maintain  $p$  within the desired bounds.

Another approach to determining  $p$  is to compute  $\bar{x}^* \bar{x}$  at each cycle and compare it with the largest previous value of  $\bar{x}^* \bar{x}$ . Then  $p$  can be chosen within the bounds determined by the previous data used in the process. This approach might permit the use of an iteration factor which is initially much larger than can be used when the factor is fixed throughout the process.

### C. COMMENTS

Neither of the preceding adaptive techniques have been applied to either real or synthetic data. A simplification of the steepest-descent algorithm, similar to the technique investigated by Bernard Widrow,<sup>\*</sup> has been investigated using time-domain data. That investigation is described in Advanced Array Research Special Report No. 1.<sup>\*\*</sup>

---

<sup>\*</sup> Widrow, Bernard, 1966, Adaptive Filters 1: Fundamentals: Stanford University Tech. Rpt. No. 6764-6, Contracts DA-01-021 AMC-90015 (Y) and NOBsr-95038, Dec.

<sup>\*\*</sup> Texas Instruments Incorporated, 1967: Adaptive Filtering of Seismic Array Data, Advanced Array Research Spec. Rpt. No. 1, Contract F33657-67-C-0708-P001, 18 Sept.



## SECTION V REQUIRED COMPUTATIONS

A major consideration in choosing a numerical method to solve a particular problem is the number of computations required by each method. This section presents a summary of the computations required by each method discussed in this report. For comparison, the method of Gaussian elimination is also included. An example is given at the conclusion of the discussion to illustrate the numerical efficiency of each method.

Each method is evaluated on the basis of solving the following equation for the elements  $a_i$  of the vector  $\bar{a}$ .

$$\left\{ \xi \begin{bmatrix} 1 & 0 & 0 & . & . & . & 0 \\ 0 & 1 & 0 & . & . & . & 0 \\ . & . & . & & & & . \\ . & . & . & & & & . \\ . & . & . & & & & . \\ 0 & . & . & . & . & . & 1 \end{bmatrix} + \begin{bmatrix} x_1 \\ . \\ . \\ . \\ . \\ x_n \end{bmatrix} \begin{bmatrix} * \\ x_1^* & . & . & . & x_n^* \end{bmatrix} \right\} \begin{bmatrix} a_1 \\ . \\ . \\ . \\ . \\ a_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix} \quad (5-1)$$

### A. GAUSSIAN ELIMINATION METHOD

By performing elementary row operations on the H matrix and the  $\bar{b}$  vector, Equation 5-1 is changed to an equation of the form

$$\begin{bmatrix} 1 & A_{12} & A_{13} & . & . & . & A_{1n} \\ 0 & 1 & A_{23} & & & & A_{2n} \\ . & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ 0 & . & . & . & . & . & A_{n-1,n} \end{bmatrix} \begin{bmatrix} a_1 \\ . \\ . \\ . \\ . \\ a_n \end{bmatrix} = \begin{bmatrix} d_1 \\ . \\ . \\ . \\ . \\ d_n \end{bmatrix} \quad (5-2)$$





The value of  $a_n$  is then  $d_n$ . The remaining elements of  $\bar{a}$  are found by successively solving for  $a_{n-1}$ ,  $a_{n-2}$ , ...,  $a_1$ . To use this method, the covariance matrix  $H$  must be computed directly. This requires  $2n$  multiplies and  $2n$  adds to calculate diagonal elements and  $2n^2 - 2n$  multiplies and  $n^2 - n$  adds to calculate the off-diagonal elements.

Then the matrix is reduced to the triangular form of Equation 5-2 which requires  $2n^3 - 2n^2$  multiplies,  $2n^3 - 2n^2$  adds, and  $2n^2$  divides. Solving for the filter weights using the triangular form (Equation 5-2) requires an additional  $2n^2 - 2n$  multiplies and  $2n^2 - 2n$  adds.

For all the computations, the Gaussian elimination method, including the covariance matrix formation, requires

- $2n^3 + 2n^2 - 2n$  multiplies
- $2n^3 + n^2 - n$  adds
- $2n^2$  divides

#### B. STEEPEST-DESCENT METHOD

Calculating the residual vector requires  $10n$  multiplies and  $8n + 1$  adds. To form the new filter weights,  $2n$  multiplies and  $2n$  adds are required. If the algorithm is cycled for  $ITER$  iterations, calculations using the steepest-descent method require

- $12n (ITER)$  multiplies
- $(10n + 1) (ITER)$  adds



### C. FINITE PROCEDURE

Assuming the imaginary part of  $p_k$  is 0, each iteration of the finite procedure uses  $10n$  multiplies and  $8n + 1$  adds to compute the residual vector. Computing the iteration constant requires  $10n + 3$  multiplies and  $8n + 2$  adds for the first iteration and  $12n + 3$  multiplies,  $10n + 2$  adds, and 1 divide for each succeeding iteration. To calculate the new filter weights,  $2n$  multiplies and  $2n$  adds at each iteration are required. If there are ITER iterations, the total computation requires

- $(24n + 3)$  (ITER) -  $2n$  multiplies
- $(20n + 3)$  (ITER) -  $2n$  adds
- ITER divides

### D. EXACT INVERSE METHOD FOR SINGLE-CHANNEL PREDICTION

Computing the filter weights by this method requires

- $3n$  multiplies
- $n + 1$  adds
- 1 divide

### E. COMPARISON

As an example, consider the design of a high-resolution  $f-k$  spectra filter set where a first approximation of the filter weights is available. Using 20 channels of data, the required amount of computation for each method is as follows:

- Gaussian Elimination — 16,760 multiplies, 16,380 adds, and 800 divides
- Steepest Descent (2 iterations) — 480 multiplies and 402 adds



- 
- Finite (1 iteration) — 443 multiplies, 363 adds, and 1 divide
  - Exact Inverse — 40 multiplies, 21 adds, and 1 divide

In this example, the exact inverse procedure is clearly the most efficient computationally. This efficiency is due to the particular formulation of the equations being solved, and the solution of a more general filter-design problem might show one of the other techniques to be more effective. The selection of a technique depends on the problem formulation and on the knowledge of the form of the solution.



---

## SECTION VI

### CONCLUSIONS

Analysis of the methods available for the numerical solution of the frequency-domain multichannel filter-design problem yields two major conclusions:

- The exact inverse equation is by far the most satisfactory method for designing high-resolution filter sets from single transform data (a rank-one matrix of data).
- The exact inverse equation can be used to update the inverse of a spectral matrix for adaptively tracking nonstationary noise fields. Such information would be required to do Bayesian location in a correlated noise field (for example, at the subarray level).



APPENDIX A  
GRADIENT METHODS



## APPENDIX A GRADIENT METHODS

A method of iteratively solving the equation  $H\bar{a} = \bar{b}$  for an unknown  $\bar{a}$  can be generated by minimizing the quadratic form

$$Q = \bar{v}^{*T} H \bar{v} - \bar{b}^{*T} \bar{v} - \bar{v}^{*T} \bar{b} \quad (A-1)$$

When  $H$  is positive-definite Hermitian, the quadratic  $\bar{v}^{*T} H \bar{v}$  is non-negative, and  $Q$  takes on its minimum value  $-\bar{b}^{*T} H^{-1} \bar{b}$  when  $\bar{v} = H^{-1} \bar{b}$ .

Thus, the vector  $\bar{v}$  which minimizes  $Q$  is the solution vector  $\bar{a}$ . The iterative procedure makes an improved estimate  $\bar{v}_k$  of the solution vector by starting with an initial estimate  $\bar{v}_0$  and successively computing a new estimate

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{w}_k \quad \text{for } p_k \text{ real} \quad (A-2)$$

The amount  $Q$  changes from the  $k$  to the  $k+1$  iteration is

$$\Delta Q = Q_{k+1} - Q_k = p_k^2 \bar{w}_k^{*T} H \bar{w}_k - p_k \left( \bar{w}_k^{*T} \bar{r}_k + \bar{r}_k^{*T} \bar{w}_k \right) \quad (A-3)$$

where

$$\bar{r}_k = \bar{b} - H \bar{v}_k$$

Minimizing  $Q$  requires that  $\Delta Q$  be negative after each iteration. This constrains  $p_k$  to the range

$$0 < p_k < \frac{\bar{w}_k^{*T} \bar{r}_k + \bar{r}_k^{*T} \bar{w}_k}{\bar{w}_k^{*T} H \bar{w}_k}$$



The largest change in  $\Delta Q$  is obtained when

$$p_k = \frac{1}{2} \left[ \frac{\bar{w}_k^* T \bar{r}_k + \bar{r}_k^* T \bar{w}_k}{\bar{w}_k^* T H \bar{w}_k} \right] \quad (A-4)$$

An optimum  $p_k$  then can be computed at each iteration. Only the choice of  $\bar{w}_k$  remains to completely define the iterative process. It is this choice of  $\bar{w}_k$  which distinguishes the various iterative methods available. By choosing  $\bar{w}_k$  to be the columns of the identity matrix taken in cyclic order, the method becomes the Gauss-Seidel technique. This technique will not be considered in this report.

#### A. STEEPEST-DESCENT METHOD

If the process is desired to move in the direction of "steepest descent," then  $\Delta Q$  is minimized with respect to the elements of  $\bar{w}_k$ . This results in choosing  $\bar{w}_k = \bar{r}_k = \bar{b} - H\bar{v}_k$ . Substituting this choice into the equation for  $p_k$  yields

$$p_k = \frac{\bar{r}_k^* T \bar{r}_k}{\bar{r}_k^* T H \bar{r}_k} \quad (A-5)$$

At this point,  $p_k$  is the reciprocal of the Rayleigh quotient for the residual vector. The Rayleigh quotient has two useful properties:

- For an arbitrary vector  $\bar{r}$ , the Rayleigh quotient always lies between the largest and the smallest eigenvalues of the matrix  $H$
- For a first-order approximation to an eigenvector of  $H$ , it yields a second-order approximation to the corresponding eigenvalue



These properties imply that only one calculation of  $p_k$  is necessary if the initial choice of  $\bar{v}_0$  is close to the desired solution  $H^{-1} \bar{b}$ . This results from the ability to express the residual vector  $\bar{r}_k$  as a linear combination of the normalized eigenvectors  $\bar{y}_\ell$  of  $H$ .

$$\bar{r}_k = \sum_{\ell=1}^n a_{\ell k} \bar{y}_\ell \quad \text{with } \bar{y}^* \bar{y} = 1 \quad (A-6)$$

The matrix  $H$  can also be expressed as the sum of its eigenvalues and eigenvectors:

$$H = \sum_{\ell=1}^n \lambda_\ell \bar{y}_\ell \bar{y}_\ell^* \quad \text{for } \lambda_i \neq \lambda_j \neq 0 \quad (A-7)$$

This simple assumption is not necessary to the conclusion, and the result is still valid when  $H$  is not of this form.

Substituting Equations A-6 and A-7 into Equation A-5 yields

$$p_k = \frac{\left[ \sum_{\ell=1}^n a_{\ell k}^* \bar{y}_\ell^* \right] \left[ \sum_{\ell=1}^n a_{\ell k} \bar{y}_\ell \right]}{\left[ \sum_{\ell=1}^n a_{\ell k}^* \bar{y}_\ell^* \right] \left[ \sum_{\ell=1}^n \lambda_\ell \bar{y}_\ell \bar{y}_\ell^* \right] \left[ \sum_{\ell=1}^n a_{\ell k} \bar{y}_\ell \right]} \quad (A-8)$$

Since orthogonal eigenvectors can be generated for a Hermitian matrix, Equation A-8 reduces to

$$p_k = \frac{\sum_{\ell=1}^n a_{\ell k}^* a_{\ell k}}{\sum_{\ell=1}^n \lambda_{\ell k} a_{\ell k}^* a_{\ell k}} \quad (A-9)$$





It is apparent here that a first-order change in the  $a_{lk}$ 's will not affect the value of  $p_k$ . If the residual vector is initially small, then  $p_k$  will be constant throughout the process.

The numerical algorithm generated using the steepest-descent approach is

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{r}_k \quad (A-10)$$

$$\bar{r}_k = \bar{b} - H \bar{v}_k \quad (A-11)$$

$$p_k = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_k^* H \bar{r}_k} \quad (A-12)$$

A simpler procedure results when the form of the H matrix is

$$H = \xi I + \sum_{l=1}^m \alpha_l \bar{x}_l \bar{x}_l^* \quad (A-13)$$

The matrix operations become vector operations, and the algorithm is

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{r}_k \quad (A-10)$$

$$\bar{r}_k = \bar{b} - \xi \bar{v}_k - \sum_{l=1}^m \alpha_l \bar{x}_l (\bar{x}_l^* \bar{v}_k) \quad (A-14)$$

$$p_k = \frac{\bar{r}_k^* \bar{r}_k}{\xi \bar{r}_k^* \bar{r}_k + \sum_{l=1}^m \alpha_l \bar{x}_l^* \bar{x}_l} \quad (A-15)$$



$$z_{lk} = \bar{x}_l^* \bar{r}_k \quad (A-16)$$

While the notation of Equations A-14, A-15, and A-16 is more complicated than that of Equations A-11 and A-12, the amount of computation needed has been reduced. The reduction in computation is obtained by performing the vector-dot-product operations  $\bar{x}_l^* \bar{v}_k$  and  $\bar{x}_l^* \bar{r}_k$  first, leaving only scalar operations to be performed. An analysis of the amount of computation required is presented in a later section of this report.

## B. FINITE PROCEDURE

By using the method of conjugate gradients, it is possible to choose  $p_k$  and  $\bar{w}_k$  so that the iteration process terminates in exactly  $n$  steps. This is accomplished by allowing  $\bar{w}_k$  to be a combination of the current residual and the previous vector  $\bar{w}_{k-1}$  so that

$$\bar{w}_k = \bar{r}_k + q_{k-1} \bar{w}_{k-1} \quad (A-17)$$

The successive constants  $q_{k-1}$  are chosen to make the quadratic form

$$\bar{w}_k^* H \bar{w}_{k-1} = 0 \quad (A-18)$$

The constant  $p_k$  is again chosen to minimize  $Q$  in Equation A-1 so that

$$p_k = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_k^* H \bar{r}_k} \quad (A-12)$$



Substituting Equation A-17 into Equation A-18 and solving for the constant  $q_{k-1}$  yields

$$q_{k-1} = - \frac{\bar{r}_k^* H \bar{w}_{k-1}}{\bar{w}_{k-1}^* H \bar{w}_{k-1}} \quad (A-19)$$

The particular choices of  $p_k$  and  $q_{k-1}$  just given produce several important effects. First, the vector product

$$\bar{r}_{k+1}^* \bar{w}_k = \bar{w}_k^* \bar{r}_{k+1} = \bar{w}_k^* (\bar{r}_k p_k H \bar{w}_k) \quad (A-20)$$

vanishes, which implies that  $\bar{r}_{k+1}$  and  $\bar{w}_k$  are orthogonal. Second, the vector product

$$\begin{aligned} \bar{r}_{k+1}^* \bar{r}_k &= (\bar{r}_k^* - p_k \bar{w}_k^* H) (\bar{w}_k - q_{k-1} \bar{w}_{k-1}) \\ &= \bar{r}_k^* \bar{w}_k - p_k \bar{w}_k^* H \bar{w}_{k-1} \end{aligned} \quad (A-21)$$

also vanishes as a result of Equation A-21 and the choice of  $p_k$ . This implies that  $\bar{r}_{k+1}$  and  $\bar{r}_k$  are also orthogonal. It can be shown by induction that

$$\bar{w}_{k+1}^* H \bar{w}_\ell = \bar{r}_{k+1}^* \bar{w}_\ell = \bar{r}_{k+1}^* \bar{r}_\ell = 0 \quad (A-22)$$

for  $\ell = 0, 1, \dots, k-1$

That is, every new  $\bar{r}_k$  computed is orthogonal to all the previously computed residual vectors. Similarly, each  $\bar{r}_k$  is orthogonal to all the previously computed  $\bar{w}_\ell$  vectors. It follows that  $\bar{r}_n = 0$ , since the  $n$  orthogonal vectors  $\bar{r}_0$  through  $\bar{r}_{n-1}$  span the  $n$ -space defined by the matrix  $H$  of rank  $n$ . These facts allow simplifying the computation of  $q_{k-1}$  to



$$q_{k-1} = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_{k-1}^* \bar{r}_{k-1}} \quad (\text{A-23})$$

The computational cycle evolving this procedure is

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{w}_k \quad (\text{A-10})$$

$$\bar{r}_k = \bar{b} - H \bar{v}_k \quad (\text{A-11})$$

$$p_k = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_k^* H \bar{r}_k} \quad (\text{A-12})$$

$$q_{k-1} = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_{k-1}^* \bar{r}_{k-1}} \quad (\text{A-23})$$

$$\bar{w}_k = \bar{r}_k + q_{k-1} \bar{w}_{k-1} \quad (\text{A-17})$$

The first cycle is computed using  $\bar{w}_0 = \bar{r}_0$ .

Again, consider the special case where

$$H = \xi I + \sum_{\ell=1}^m a_{\ell} \bar{x}_{\ell} \bar{x}_{\ell}^* \quad (\text{A-13})$$



The computational cycle becomes

$$\bar{v}_{k+1} = \bar{v}_k + p_k \bar{w}_k \quad (\text{A-10})$$

$$\bar{r}_k = \bar{b} - \xi \bar{v}_k - \sum_{\ell=1}^m \alpha_{\ell} \bar{x}_{\ell} (\bar{x}_{\ell}^* \bar{v}_k) \quad (\text{A-14})$$

$$u_k = \bar{r}_k^* \bar{r}_k \quad (\text{A-24})$$

$$z_{\ell k} = \bar{x}_{\ell}^* \bar{r}_k \quad (\text{A-16})$$

$$F_k = u_k / \left( \xi u_k + \sum_{\ell=1}^m \alpha_{\ell} z_{\ell k}^* z_{\ell k} \right) \quad (\text{A-25})$$

$$q_{k-1} = u_k / u_{k-1} \quad (\text{A-26})$$

$$\bar{w}_k = \bar{r}_k + q_{k-1} \bar{w}_{k-1} \quad (\text{A-17})$$

For the first iteration,  $\bar{w}_0 = \bar{r}_0$  and  $q_{k-1}$  is not computed.

Notation again becomes more cumbersome, but the computation needed is much less than that required to form the autopower-crosspower matrix and then solve for the desired filter set. Computational savings are obtained by carrying out the same computations in a different order.



APPENDIX B  
EXACT INVERSE EQUATION



## APPENDIX B

### EXACT INVERSE EQUATION

#### A. SOLUTION BY THE EXACT INVERSE EQUATION

The exact inverse equation is an analytic expression for the inverse of the matrix  $H$  in the equation

$$H \bar{a} = \bar{b} \quad (B-1)$$

The solution vector  $\bar{a}$  can be written as

$$\bar{a} = H^{-1} \bar{b} \quad (B-2)$$

when  $H$  is nonsingular. This report studies numerical methods of solving Equation B-1 when the  $H$  matrix is

$$H = \xi S + \sum_{l=1}^m \alpha_l \bar{x}_l \bar{x}_l^* \quad (B-3)$$

Consider a matrix of the form  $[A + \bar{x} \bar{y}^*]$ . When both  $A$  and  $[A + \bar{x} \bar{y}^*]$  are nonsingular, the inverse of this matrix is

$$[A + \bar{x} \bar{y}^*]^{-1} = A^{-1} - \frac{A^{-1} \bar{x} \bar{y}^* A^{-1}}{1 + \bar{y}^* A^{-1} \bar{x}} \quad (B-4)$$

where

$A$  =  $n \times n$  complex matrix

$\bar{x}$  =  $n \times 1$  column vector

$\bar{y}^*$  =  $1 \times n$  row vector



This equation is verified by the following multiplication:

$$\left[ A^{-1} - \frac{A^{-1} \bar{x} \bar{y}^* A^{-1}}{(1 + \bar{y}^* A^{-1} \bar{x})} \right] [A + \bar{x} \bar{y}^*] \quad (B-5)$$

$$= A^{-1} A + A^{-1} \bar{x} \bar{y}^* - \frac{A^{-1} \bar{x} \bar{y}^* A^{-1} A}{(1 + \bar{y}^* A^{-1} \bar{x})} - \frac{A^{-1} \bar{x} \bar{y}^* A^{-1} \bar{x} \bar{y}^*}{(1 + \bar{y}^* A^{-1} \bar{x})}$$

$$= I + A^{-1} \bar{x} \bar{y}^* - \frac{A^{-1} \bar{x} \bar{y}^* - A^{-1} \bar{x} \left( \frac{\bar{y}^* A^{-1} \bar{x}}{\bar{y}} \right) \bar{y}^*}{(1 + \bar{y}^* A^{-1} \bar{x})}$$

$$= I + \frac{\left( 1 + \bar{y}^* A^{-1} \bar{x} \right) A^{-1} \bar{x} \bar{y}^* - A^{-1} \bar{x} \bar{y}^* - \left( \bar{y}^* A^{-1} \bar{x} \right) A^{-1} \bar{x} \bar{y}^*}{(1 + \bar{y}^* A^{-1} \bar{x})}$$

$$= I$$

When the matrix H has the form of Equation B-3, a recursive scheme can be used to generate the inverse of H directly from the transform points. The exact inverse procedure finds

$$\left[ \xi S + \sum_{\ell=1}^m \alpha_{\ell} \bar{x}_{\ell} \bar{x}_{\ell}^* \right]^{-1} = H^{-1} \quad (B-6)$$





Equation B-6 is found by first computing

$$A_1^{-1} = \left[ \xi S + \alpha_1 \bar{x}_1 \bar{x}_1^* \right]^{-1} \quad (B-7)$$

$$= \frac{1}{\xi} \left[ S^{-1} - \left( \frac{\alpha_1}{\xi + \alpha_1 \bar{x}_1^* S^{-1} \bar{x}_1} \right) S^{-1} \bar{x}_1 \bar{x}_1^* S^{-1} \right]$$

$$A_2^{-1} = \left[ A_1^{-1} - \alpha_2 \bar{x}_2 \bar{x}_2^* \right]^{-1}$$

$$A_2^{-1} = A_1^{-1} - \left( \frac{\alpha_2}{1 + \alpha_2 \bar{x}_2^* A_1^{-1} \bar{x}_2} \right) A_1^{-1} \bar{x}_2 \bar{x}_2^* A_1^{-1}$$

$$\vdots$$

$$A_m^{-1} = A_{m-1}^{-1} - \left( \frac{\alpha_{m-1}}{1 + \alpha_{m-1} \bar{x}_m^* A_{m-1}^{-1} \bar{x}_m} \right) A_{m-1}^{-1} \bar{x}_m \bar{x}_m^* A_{m-1}^{-1}$$

and

$$H^{-1} = A_m^{-1} \quad (B-8)$$

Each iteration is completed by calculating the vector

$$\bar{c}_l = A_{l-1}^{-1} \bar{x}_l \quad (B-9)$$

Then the vector  $\bar{d}$  has the form

$$\bar{d}_l = \alpha_l \bar{c}_l \quad (B-10)$$



The first iteration requires a knowledge of the matrix  $S^{-1}$  and the scalar  $\xi$  which are stored in the computer memory or calculated at the beginning of each cycle. Then, from Equation B-8,

$$A_1^{-1} = \frac{1}{\xi} S^{-1} - \left( \frac{\alpha_1}{\xi^2 + \xi \alpha_1 \bar{x}_1^* S^{-1} \bar{x}} \right) S^{-1} \bar{x}_1 \bar{x}_1^* S^{-1} \quad (B-11)$$

Each succeeding iteration is

$$A_{l+1}^{-1} = A_l^{-1} - \left( \frac{\alpha_l}{1 + \bar{d}_l^* \bar{x}_l} \right) \bar{c} \bar{d}^* \quad \text{for } l = 1, 2, \dots, m-1 \quad (B-12)$$

Since the  $\bar{x}_l$ 's are arbitrary complex vectors, this algorithm could be used either to stack a series of autopower-crosspower matrices from successive time gates or to stack adjacent frequency points (smoothing) from the same time gate.

#### B. HIGH-RESOLUTION f-k FILTER SETS

The high-resolution technique uses an estimate of the auto-crosspower matrix of the form

$$H = \xi I + \bar{z} \bar{z}^* \quad (B-13)$$

where

$I$  =  $n \times n$  identity matrix

$\bar{z}$  =  $n \times 1$  column vector

$\bar{z}^*$  = conjugate transpose of  $\bar{z}$

$\xi$  = real constant



The elements of the vector  $\bar{z}$  are the normalized elements of the data transform vector  $\bar{x}$ ; i. e.,

$$z_i = \bar{x}_i / (\bar{x}_i \bar{x}^*)^{1/2} \quad \text{for } i = 1, \dots, n \quad (\text{B-14})$$

Using the exact inverse equation, the inverse of the H matrix is

$$H^{-1} = \frac{1}{\xi} \left[ I - \frac{\bar{z} \bar{z}^*}{\xi + \bar{z}^* \bar{z}} \right] \quad (\text{B-15})$$

Multiplying  $\bar{b}$  by  $H^{-1}$  yields a filter set

$$\bar{a} = \frac{1}{\xi} \bar{b} - \frac{\bar{z} \bar{z}^* \bar{b}}{(\xi + \bar{z}^* \bar{z})} \quad (\text{B-16})$$

Further simplification is obtained when  $\bar{z}^* \bar{z} = N$ . The filter set becomes

$$\bar{a} = \frac{1}{\xi} \bar{b} - \left( \frac{1}{\xi} \right) \left( \frac{1}{\xi + n} \right) \bar{z} \bar{z}^* \bar{b} \quad (\text{B-17})$$

Since the  $\bar{b}$  vector has one nonzero element in the  $j^{\text{th}}$  value, Equation B-17 is computed more simply as

$$\begin{aligned} a_l &= \frac{1}{\xi} \left( \frac{1}{\xi + n} \right) z_j^* z_l \quad l = 1, 2, \dots, n \\ a_j &= \frac{1}{\xi} \left[ 1 - \left( \frac{1}{\xi + n} \right) z_j^* z_j \right] \quad l \neq j \end{aligned} \quad (\text{B-18})$$

The computational efficiency of this method over all the other methods discussed is obvious. The filter sets designed by each technique were virtually identical.



APPENDIX C  
NUMERICAL RESULTS

**BLANK PAGE**



## APPENDIX C

### NUMERICAL RESULTS

#### A. STEEPEST-DESCENT METHOD

The numerical techniques discussed previously have been applied to the special problem of designing high-resolution  $f$ - $\vec{k}$  spectra filter sets. In this special situation, the techniques can be made more simple to achieve even greater numerical efficiency.

The high-resolution technique uses an estimate of the auto-crosspower matrix of the form

$$H = \xi I + \bar{z} \bar{z}^* \quad (C-1)$$

where

$I$  =  $n \times n$  identity matrix

$\bar{z}$  =  $n \times 1$  column vector

$\bar{z}^*$  = conjugate transpose of  $\bar{z}$

$\xi$  = real constant

The elements of the vector  $\bar{z}$  are the normalized elements of the data transform vector  $\bar{x}$ ; i. e.,

$$z_i = \bar{x}_i / \left( \bar{x}_i \bar{x}_i^* \right)^{1/2} \quad \text{for } i = 1, \dots, n \quad (C-2)$$

In this situation, the residual vector  $\bar{r}_k$  is

$$\bar{r}_k = \bar{b} - \left[ \xi I + \bar{z} \bar{z}^* \right] \bar{v}_k \quad (C-3)$$



The iteration factor  $p_k$  is not affected by a first-order change in the residual vector  $\bar{r}_k$ . Further observations about the iteration factor can be made as a result of the special form of the H matrix being considered.

It is desired to compute the iteration factor

$$p_k = \frac{\bar{r}_k^* \bar{r}_k}{\bar{r}_k^* H \bar{r}_k} \quad (C-4)$$

The related eigenvalue problem is

$$H \bar{y}_l = \lambda_l \bar{y}_l \quad \text{for } \bar{y}_l^* \bar{y}_l = 1 \quad (C-5)$$
$$l = 1, 2, \dots, n$$

where

$\lambda_l$  = eigenvalue of H

$\bar{y}_l$  = the related eigenvector

When H is  $(\xi I + \bar{x} \bar{x}^*)$ , it has only one eigenvalue  $\lambda_n$ ,

$$\lambda_n = \xi + \bar{x}^* \bar{x} \quad (C-6)$$

and all the n-1 other eigenvalues are  $\xi$ .

Suppose an initial vector  $\bar{v}_1$  is chosen so that

$$\bar{r}_1 = \alpha \bar{y}_n + \bar{\epsilon} \quad (C-7)$$



where  $\bar{y}_n$  is the eigenvector corresponding to the nonzero eigenvalue  $\lambda_n$  and  $\bar{\epsilon}$  is a vector representing the difference between  $\bar{r}_1$  and  $\alpha \bar{y}_n$ . Then,

$$p_1 = \frac{[\alpha \bar{y}_n^* + \bar{\epsilon}^*][\alpha \bar{y}_n + \bar{\epsilon}]}{[\alpha \bar{y}_n^* + \bar{\epsilon}^*] H [\alpha \bar{y}_n + \bar{\epsilon}]} \quad (C-8)$$

$$= \frac{\alpha^2 \bar{y}_n^* \bar{y}_n + \alpha [\bar{y}_n^* \bar{\epsilon} + \bar{\epsilon}^* \bar{y}_n] + \bar{\epsilon}^* \bar{\epsilon}}{\alpha^2 \bar{y}_n^* H \bar{y}_n + \alpha [\bar{y}_n^* H \bar{\epsilon} + \bar{\epsilon}^* H \bar{y}_n] + \bar{\epsilon}^* H \bar{\epsilon}}$$

By replacing  $H \bar{y}_n$  with  $\lambda_n \bar{y}_n$  in the denominator of Equation C-8 and recognizing that  $\bar{y}_n^* \bar{y}_n = 1$  by definition, it is possible to write

$$p_1 = \frac{\alpha^2 + \alpha (\bar{y}_n^* \bar{\epsilon} + \bar{\epsilon}^* \bar{y}_n) + \bar{\epsilon}^* \bar{\epsilon}}{\lambda_n [\alpha^2 + \alpha (\bar{y}_n^* \bar{\epsilon} + \bar{\epsilon}^* \bar{y}_n)] + \bar{\epsilon}^* H \bar{\epsilon}} \quad (C-9)$$

If the difference vector  $\bar{\epsilon}$  is small compared to the eigenvector  $\bar{y}_n$ , then Equation C-9 can be approximated as

$$p_1 = \frac{1}{\lambda_n} \quad (C-10)$$

Since

$$\lambda_n = \xi + \bar{x}^* \bar{x}$$

then,

$$p_1 = \left( \frac{1}{\xi + \bar{x}^* \bar{x}} \right) \quad (C-11)$$





When the H matrix has the form in Equation C-1,

$$p_1 = \frac{1}{\xi + \frac{\bar{z}^* \bar{z}}{n}} = \frac{1}{\xi + n} \quad (C-12)$$

where  $n$  is the number of elements in the vector  $\bar{z}$ . The value of  $p_1$  is observed in Figure C-1.

The objective then is to find an initial vector  $\bar{v}_1$  which will result in a small  $\bar{e}$  component in the first residual vector. If this is found, the iteration factor need not be computed at all but can be taken to be the value of Equation C-12.

High-resolution  $f$ - $k$  spectra are generated from a filter set  $\bar{a}$  designed using an H matrix in the form of Equation C-1 and

$$\bar{b}^* = [1, 0, 0, \dots, 0] \quad (C-13)$$

The unit entry in  $\bar{b}$  is placed in the position corresponding to the desired reference sensor. All other elements of the  $\bar{b}$  vector are 0. The equation set being solved is then

$$\left\{ \xi \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ & & \ddots \\ 0 & 1 & 1 \end{bmatrix} + \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \begin{bmatrix} z_1^* & \dots & z_n^* \end{bmatrix} \right\} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (C-14)$$

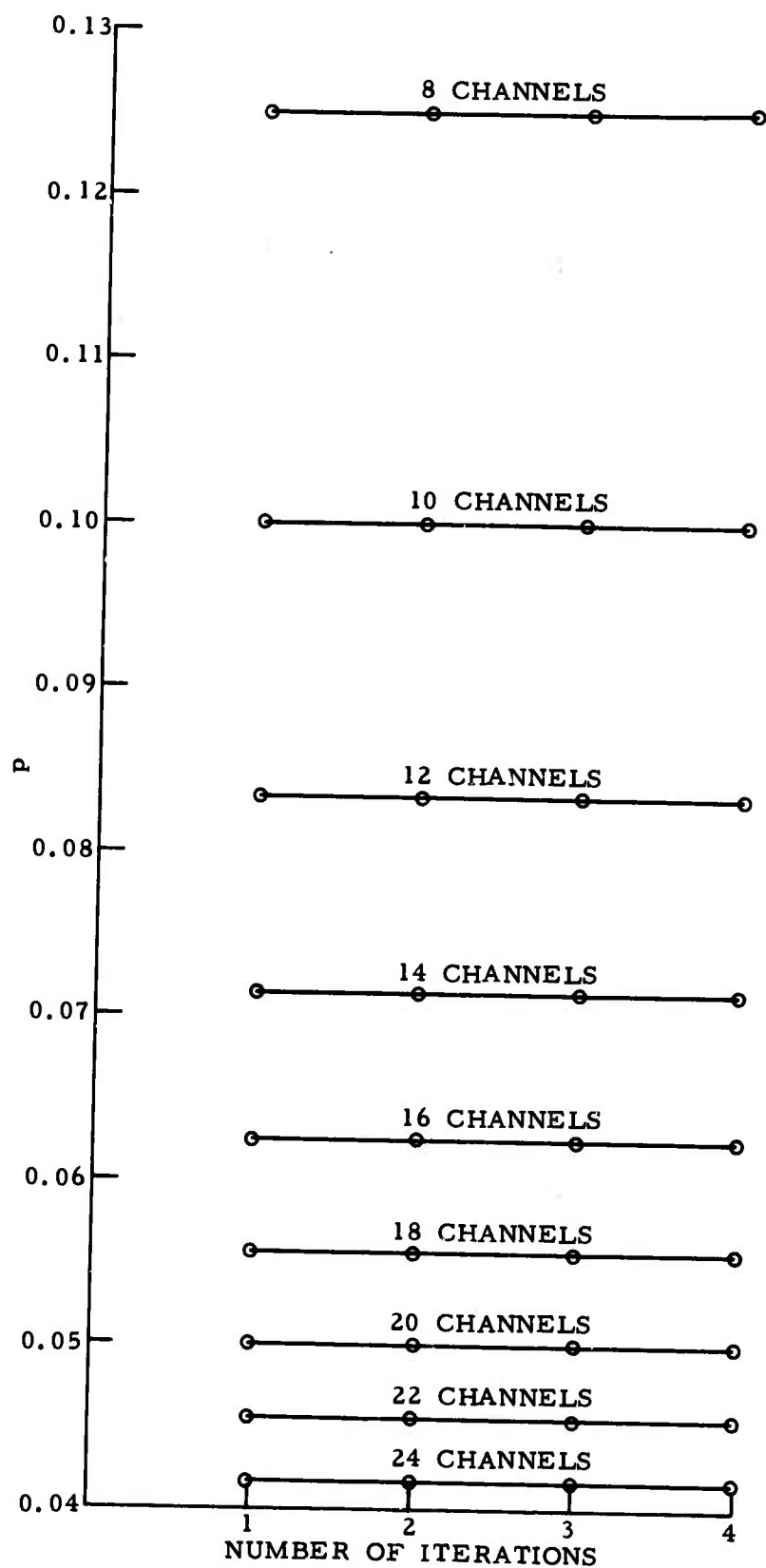


Figure C-1. Variation of  $p_1$  from Iteration to Iteration when the Residual Is Small (Real Data;  $\xi = 0.001$ )



Experimental results show that  $\bar{v}_1 = \bar{b}$  is a good choice for the initial "guess" at  $\bar{a}$ . Using this value for  $\bar{v}_1$ , Equation C-12 to solve Equation C-14, and the steepest-descent method, an accurate answer often was reached in one iteration; seldom were more than three iterations required. Bodewig\* has mentioned that  $0.9 p_1$  sometimes produces better convergence than  $p_1$ . On real data,  $0.9 p_1$  is not so good a choice as  $p_1$ . The convergence rates using different iteration factors can be observed in the Figure C-2 plots for each iteration of error vs iteration constants. When synthetic data were used, the  $0.9 p_1$  iteration factor converged faster than  $p_1$  (Figure C-3).

If a good approximation to the filter weights is not available, very slow convergence can result. This is demonstrated in Figure C-4 for synthetic data with an initial vector  $\bar{v}_1$  value of  $(1 + j1)$  for every entry where  $j = \sqrt{-1}$ .

There is an indication that solving the degenerate problem caused the rapid convergence experienced using a steepest-descent design procedure. An initially good "guess" at the filter weights is not available when the H matrix contains the sum of many vector products, implying that, in general, relatively slow convergence can be expected in the more complicated problems.

## B. FINITE PROCEDURE

Results also have been good using the finite procedure. The chief advantage of this method — that it is not necessary to start from a first approximation to the filter weights as was necessary when using the steepest-descent method — is shown in Figure C-5.

---

\* Bodewig, E., 1956: Matrix Calculus, Interscience Publishers, Inc., New York.

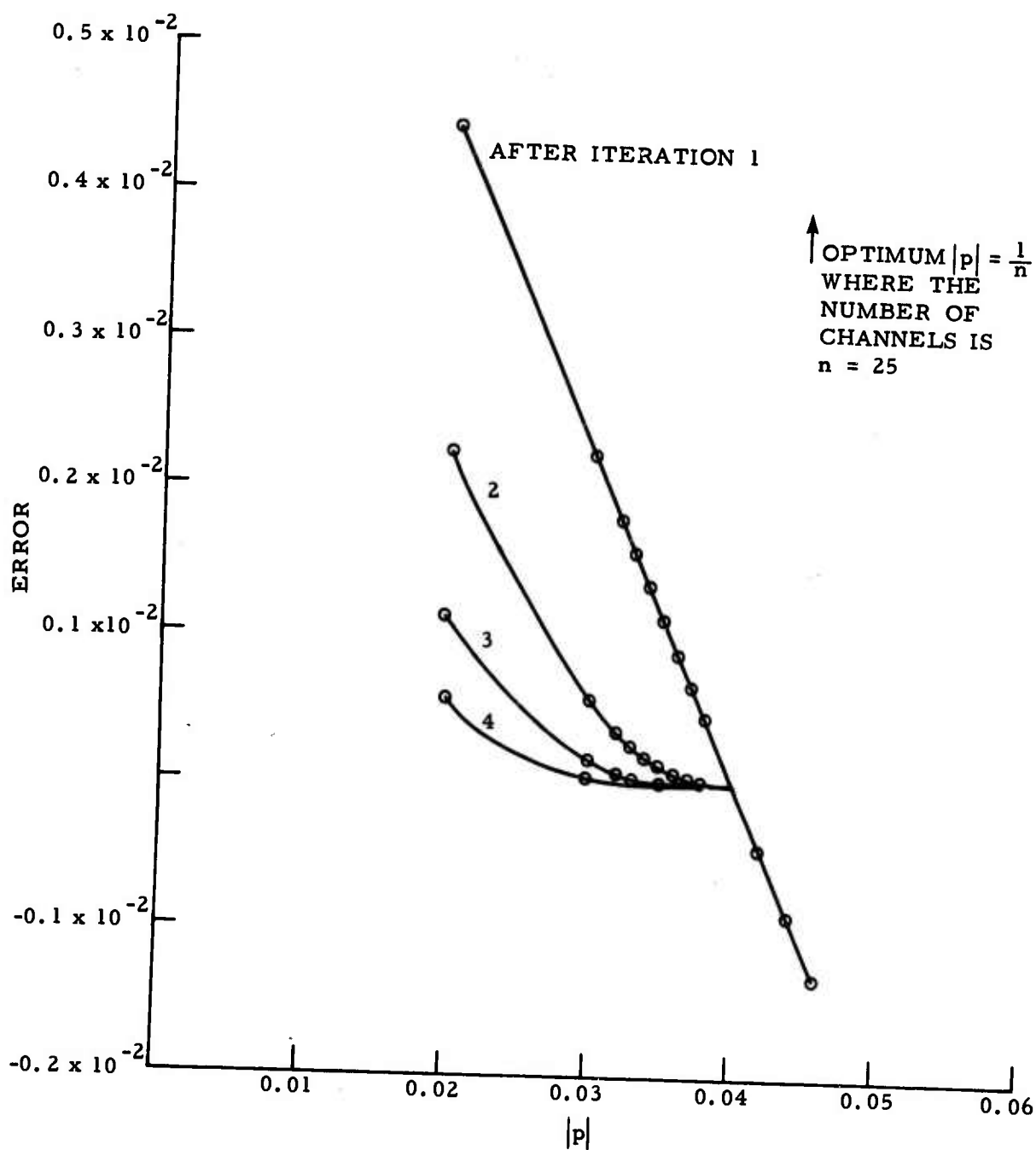


Figure C-2. Effect of Using a  $p_1$  Which Is Slightly Different from  $|p_{opt}| = 1/n$  (Steepest descent; real data;  $\xi = 0.001$ )

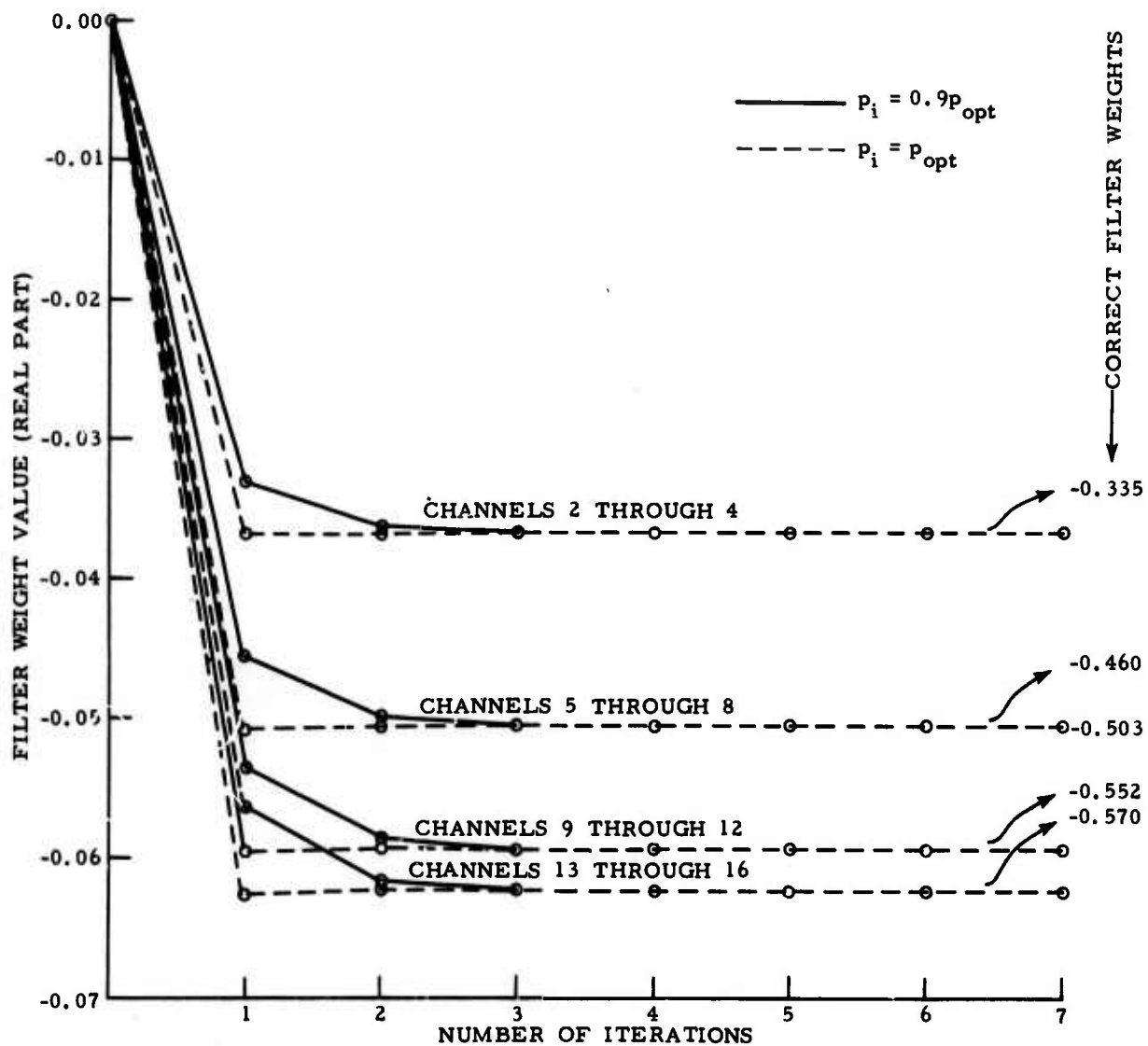


Figure C-3. Approximate Convergence of Filter Weights When Synthetic Data Are Used. (Plane-wave incident on 16-channel square array; channel 1 is reference;  $\xi = 0.1$ )

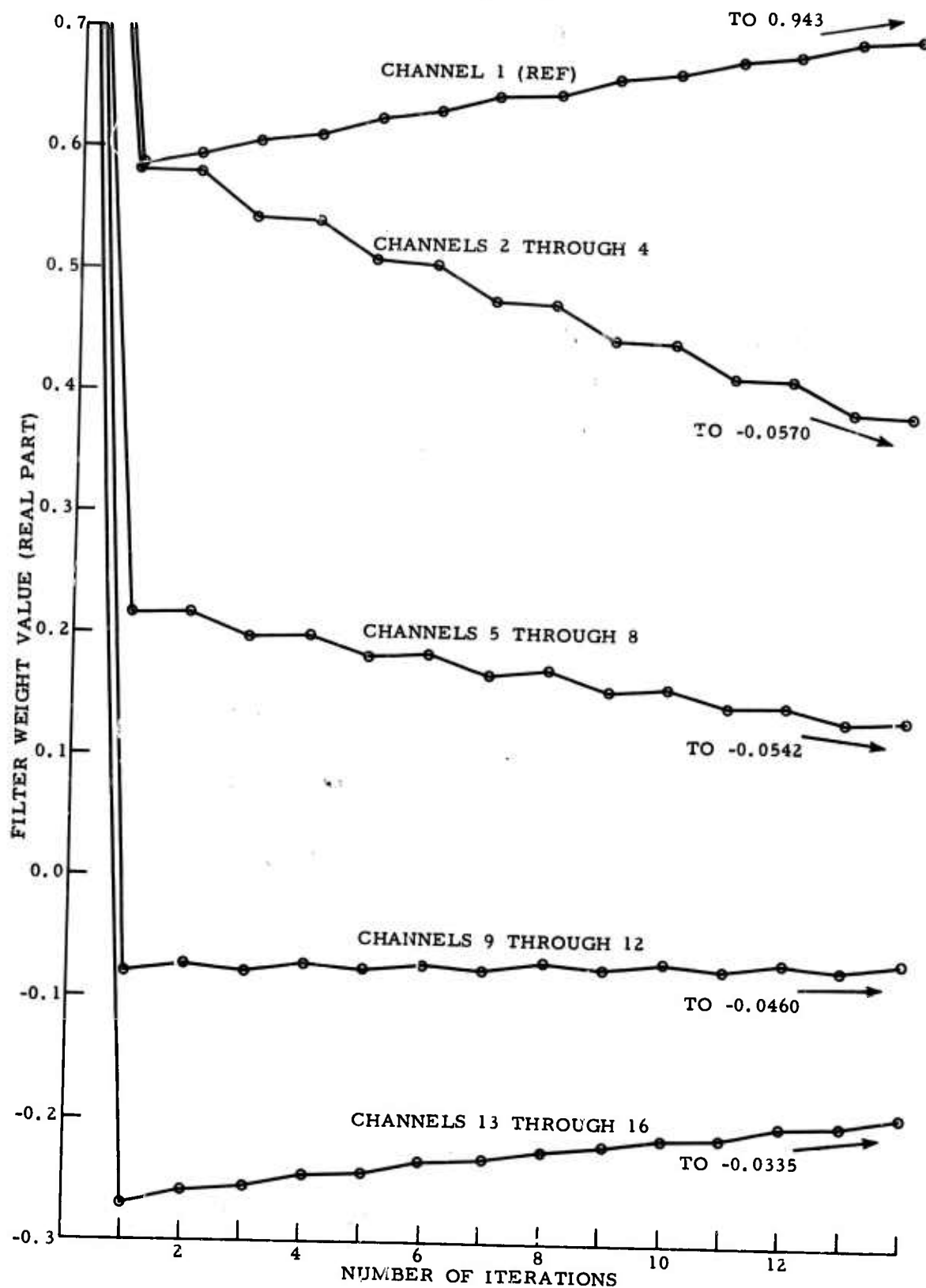


Figure C-4. Slow Convergence of Filter Weights When All Initial Filter Weights Are 1 (Synthetic data;  $\xi = 0.1$ )

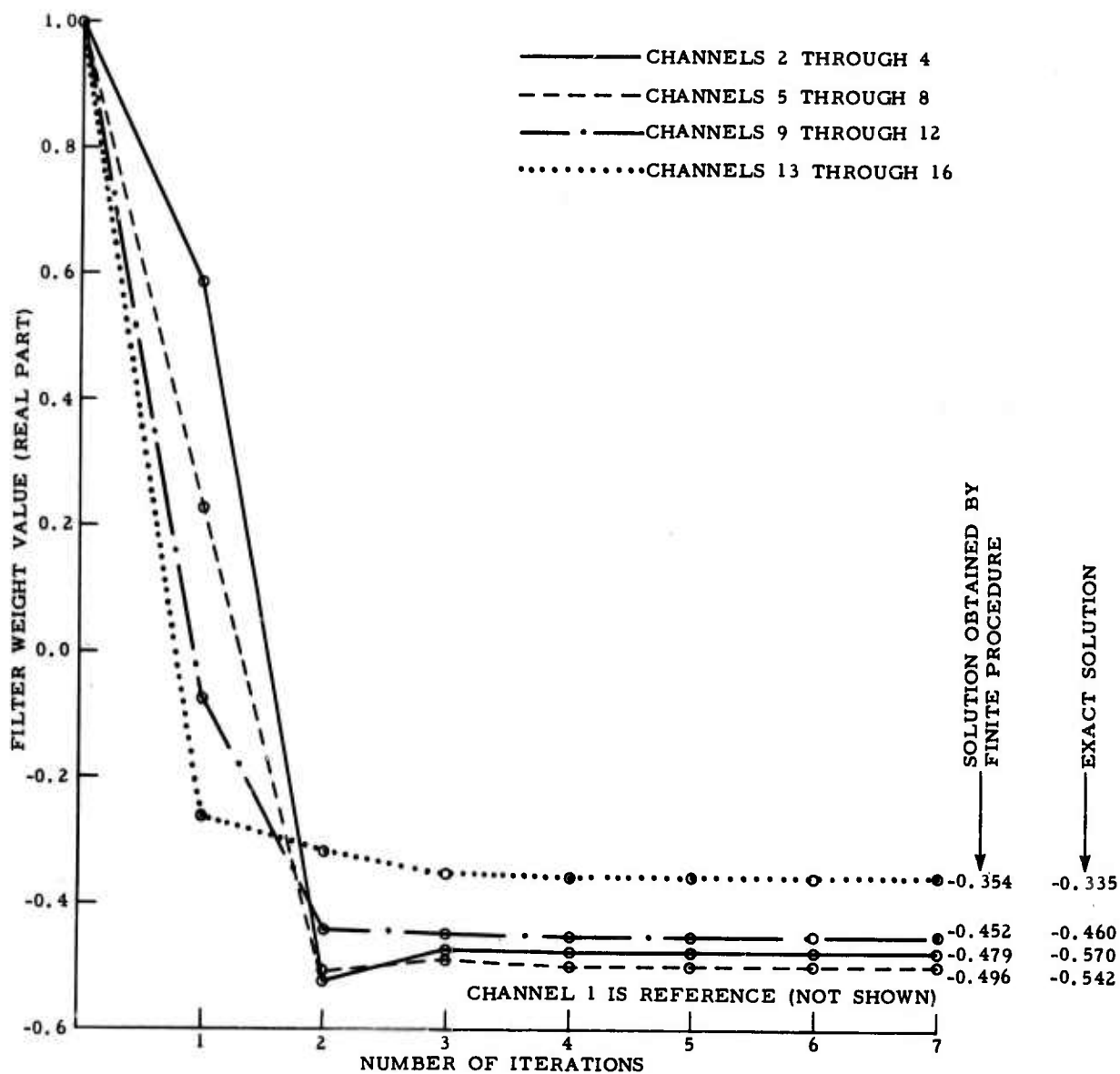


Figure C-5. Approximate Convergence of Filter Weights When All Weights Are Initially Set Equal to 1. [Finite procedure; synthetic data;  $\xi = 0.1$ ;  $(\xi I + x x^*) \bar{a} = \bar{b}$ ]



Using real data, convergence to essentially the exact solution was obtained in two iterations; every value of the initial filters was 1 (Figure C-6). Of course, when the real part of the reference channel was set to 1 and all other weights set equal to 0, much better convergence occurred (Figure C-7). In fact, the filters were correct to four significant figures after the first iteration.

Since the imaginary part of  $p_k$  was very small during the first few iterations (Table C-1), the program was rewritten to set the imaginary part to 0, thereby reducing the number of computations. The convergence was completely unaffected.

Finally,  $p_k = 0.9 p_1$  was used to measure the effect of varying  $p_k$ . Filter weights diverged rapidly (Figure C-8); the strong susceptibility to changes in  $p$  indicated that  $p$  necessarily must be recalculated at each iteration, especially in view of the large and apparently unpredictable (without recalculating) variations in  $p_k$  (Table C-1).

Although the number of computations involved is almost twice that required by the steepest-descent method, the finite procedure is better than any direct method other than the exact inverse equation. The rapid convergence of this method is certainly the result of the degenerate problem being considered.

### C. SPECTRA

Wavenumber spectra obtained from the iteration schemes presented compare quite favorably with those obtained from current high-resolution techniques. Spectra were obtained from real data (C3 subarray, minimum velocity 10.0 km/sec,  $f = 0.85$  Hz) using filter weights designed by the exact inverse equation and by the finite procedure. In both cases,



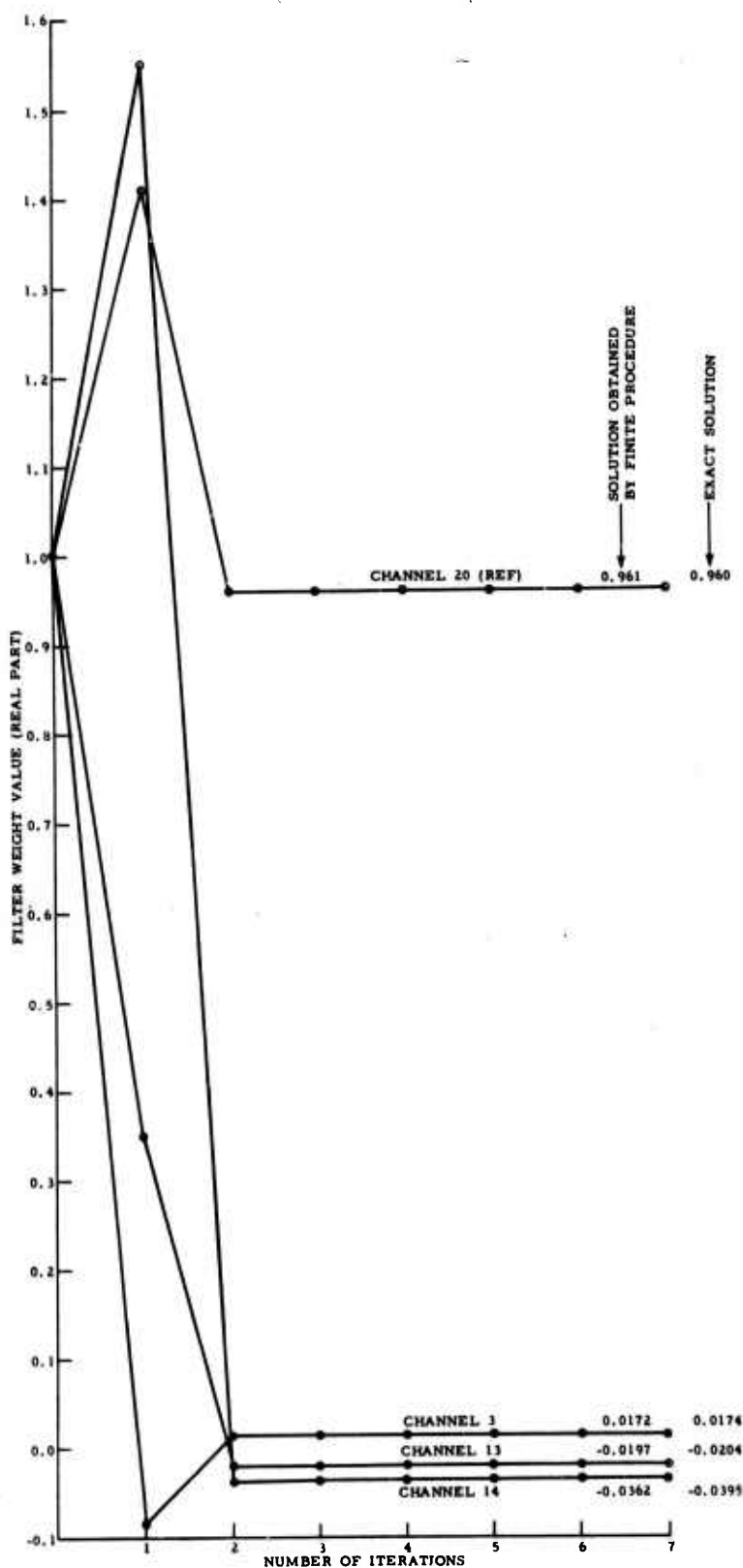


Figure C-6. Convergence of Filter Weights When All Weights are Initially Set Equal to 1 (Finite procedure; real data;  $\xi = 0.001$ )

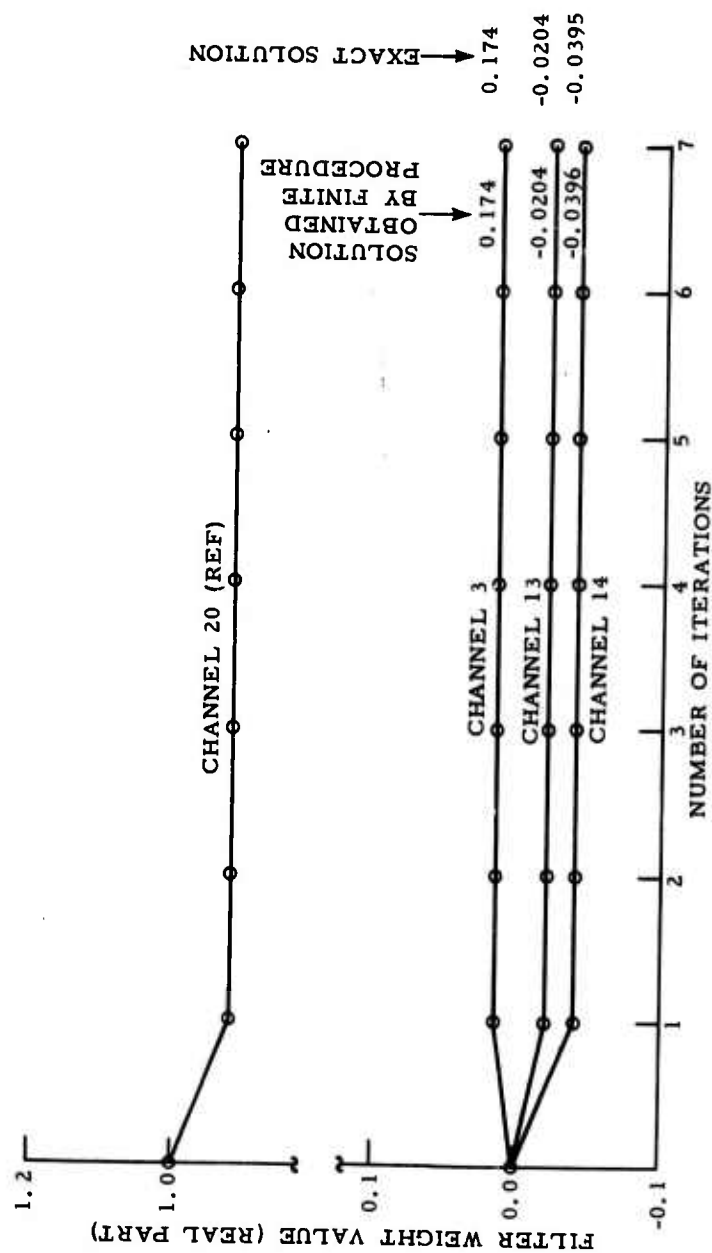


Figure C-7. Convergence of Filter Weights When Real Part of Reference Channel Is Set Equal to 1 (Finite procedure; real data  $\xi = 0.001$ )

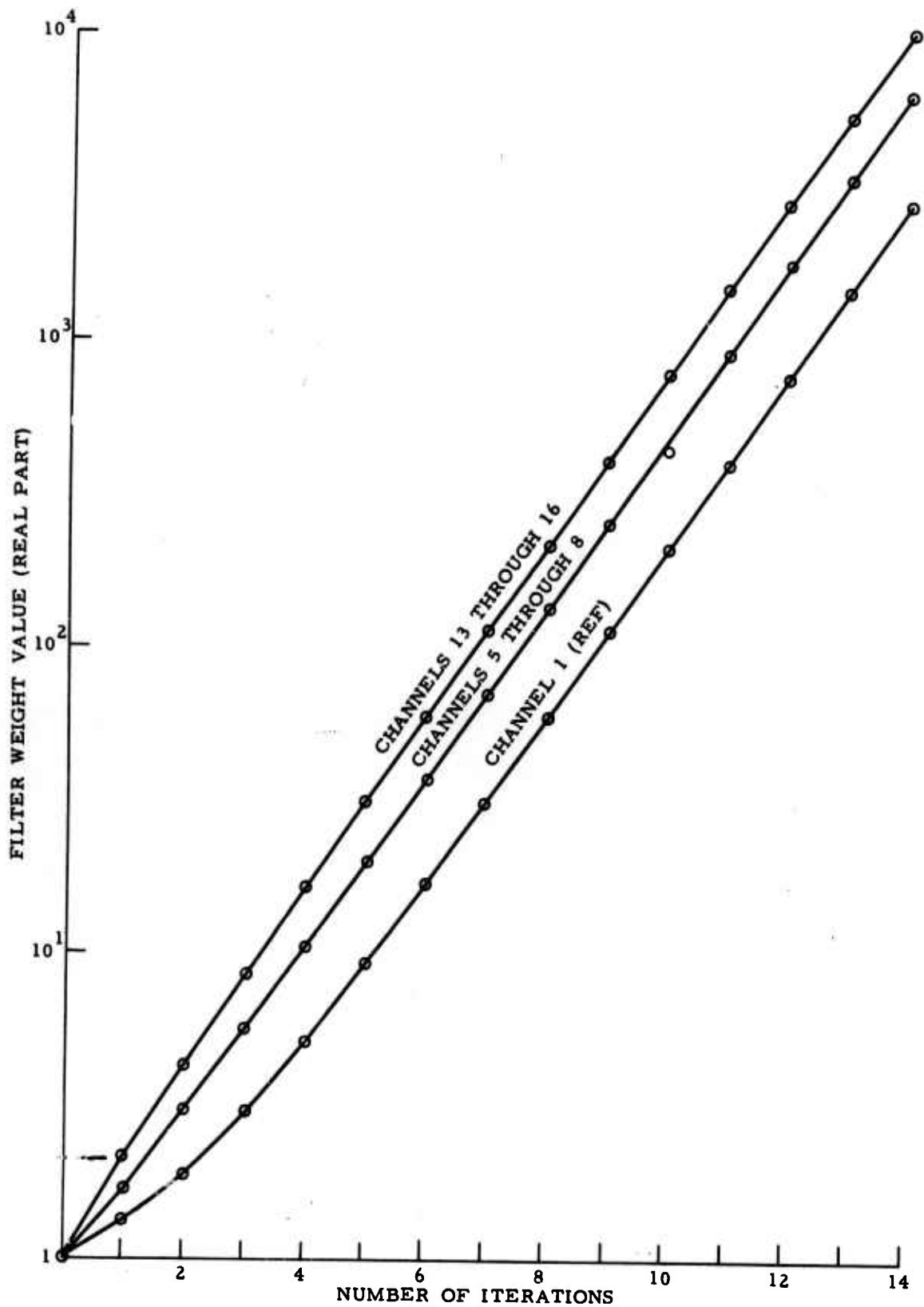


Figure C-8. Divergence of Filter Weights When  $p_1$  Is Calculated as  $p_k$   
(Synthetic data,  $\xi = 0.1$ )



Table C-1  
VARIATION OF  $p$  IN FINITE PROCEDURE (REAL DATA)

Iteration	$p$ (real part)	$p$ (imaginary part)
1	-0.0400	0.
2	-0.0400	$2.50 \times 10^{-10}$
3	-0.0400	$7.54 \times 10^{-6}$
4	-0.0278	0.0346
5	-0.0248	0.0104

Note: Initially, the real part of the reference channel is 1 and all other channels are 0.

Iteration	$p$ (real part)	$p$ (imaginary part)
1	-0.0400	0.
2	-999.	$-2.76 \times 10^{-6}$
3	-0.0399	$-9.63 \times 10^{-10}$
4	-0.0417	$-4.31 \times 10^{-8}$
5	-0.0244	$-4.78 \times 10^{-4}$
6	-0.0109	$-1.83 \times 10^{-6}$

Note: Initially, both real and imaginary parts of all filter weights are 1.



the filter weight of channel 8 has a large error because its transform was punched incorrectly on the data card. However, the spectra obtained are virtually identical with those obtained from current techniques. Since the error in channel 8 did not significantly affect the spectra, wavenumber spectra are seen to be relatively insensitive to errors in the individual filter weights (unless the error happens to be in the reference channel).

Filter weights designed from the steepest-descent method were virtually identical to those obtained from the exact inverse equation; therefore, the steepest-descent method also will yield good spectra.

#### D. SMOOTHING

Fairly good results were obtained when data were smoothed over a range of frequencies. The data used were derived from the data for the first eight channels of the real data from the C3 LASA subarray. The transform for a given channel was altered to give four new transforms with an average equal to that of the original transform. The deviation of any given component from that of the transform from which it was derived was 1 to 2 percent. If these deviations were truly random, the smoothed filter weights would be expected to be equal to those for the original data. Since the deviations were not entirely random, however, they were actually somewhat different.

With all initial filter weights set to 0 except for the real part of the reference channel which was set to 1, the steepest-descent method showed a large initial correction followed by very slow convergence. This method, therefore, seems to be somewhat ineffective.

The finite procedure, on the other hand, produced convergence to a nonvarying set of filter weights after three iterations and was reasonably correct (two significant figures) after two. These results occurred only when the initial filter set was a first approximation to the correct set of weights.



When a bad first approximation was made (e. g., letting the imaginary part of the reference channel be set to 1), very slow convergence followed a large initial correction.

Thus, the finite procedure produces good results on full rank matrices if a reasonable approximation to the filter weights is available. The steepest-descent method does not produce good results unless a good estimate of the filter weights is available.

The necessity of a good first approximation is further illustrated when data from the 10-channel CPO array were smoothed over a range of frequencies; (only nine channels were used). The data sample consisted of noise at 1.00 Hz for 16 October 1964, with 41 Fourier transforms taken from  $f = 0.95$  to 1.05 Hz at frequency increments of 0.0025 Hz. All transforms were equally weighted. Neither method could produce convergence when the real part of the reference channel was 1.0 or 0.1. Table C-2 shows the filter weights actually obtained using other methods. It is readily apparent that the initial choices did not approximate these filter weights and also that no general way to make a good approximation seems apparent. It should be noted that very poor resolution was obtained with the filter weights of Table C-2, indicating that the data were very poor.

Effective iterative methods for smoothing are thus limited to cases in which the transforms at the various frequencies under consideration are approximately equal, allowing all initial filter weights to be approximated as 0 (except the reference channel which is 1). It appears that the finite procedure is preferable to the steepest-descent method when computing spectra for smoothed crosspower matrices.



Table C-2  
FILTER WEIGHTS FOR DATA SMOOTHED OVER 41 FREQUENCIES  
FROM 0.95 TO 1.05 Hz

<u>Channel</u>	<u>Filter Weights</u>	
1	$0.22481 \times 10^{-10}$	$-0.71687 \times 10^{-7}$
2	$-0.41205 \times 10^{-2}$	$0.16657 \times 10^{-1}$
3	$0.29080 \times 10^{-2}$	$0.5448 \times 10^{-2}$
4	$-0.12049 \times 10^{-2}$	$0.16102 \times 10^{-2}$
5	$-0.10966 \times 10^{-1}$	$-0.13375 \times 10^{-3}$
6	$-0.78141 \times 10^{-3}$	$0.84145 \times 10^{-4}$
7	$0.14035 \times 10^{-1}$	$0.40036 \times 10^{-2}$
8	$-0.96124 \times 10^{-2}$	$-0.75366 \times 10^{-2}$
9	$0.22133 \times 10^{-2}$	$-0.96892 \times 10^{-2}$

UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

## 1. ORIGINATING ACTIVITY (Corporate author)

Texas Instruments Incorporated  
Science Services Division  
P.O. Box 5621, Dallas, Texas 75222

## 2. REPORT SECURITY CLASSIFICATION

Unclassified

## 2b. GROUP

--

## 3. REPORT TITLE

LARGE-ARRAY SIGNAL AND NOISE ANALYSIS — SPECIAL  
SCIENTIFIC REPORT NO. 16 — ITERATIVE TECHNIQUES FOR  
THE SOLUTION OF FREQUENCY-DOMAIN FILTER SETS

## 4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Special Scientific

## 5. AUTHOR(S)

Fenner, Peter R.

## 6. REPORT DATE

22 March 1968

## 7a. TOTAL NO. OF PAGES

57

## 7b. NO. OF REFS

5

## 8a. CONTRACT OR GRANT NO.

Contract No. AF 33(657)-16678

## b. PROJECT NO.

AFTAC Project No. VT/6707

## c.

## 9a. ORIGINATOR'S REPORT NUMBER(S)

## 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

## 10. DISTRIBUTION STATEMENT

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Chief, AFTAC.

## 11. SUPPLEMENTARY NOTES

ARPA Order No. 599

## 12. SPONSORING MILITARY ACTIVITY

Air Force Technical Applications Center  
VELA Seismological Center  
Headquarters, USAF, Washington, D. C.

## 13. ABSTRACT

Computation of high-resolution wavenumber spectra includes the solution of a set of Hermitian equations. This set of equations has the form of a least-squares multichannel frequency-domain filter design equation which can be expressed in vector-matrix notation as  $H\bar{a} = \bar{b}$ , where  $H = n \times n$  nonsingular Hermitian power spectral matrix;  $\bar{b} = n \times 1$  known complex column vector of output power spectra; and  $\bar{a} = n \times 1$  unknown complex column vector of filter weights. This report investigates three techniques of solving for the unknown vector  $\bar{a}$ : the method of conjugate gradients, steepest-descent method, and exact-inverse method. The object is to determine the accuracy and computational complexity of each technique.

Analysis of the methods available for the numerical solution of the frequency-domain multichannel filter-design problem yields two major conclusions. The exact inverse equation is by far the most satisfactory method for designing high-resolution filter sets from single transform data (a rank-one matrix of data). The exact inverse equation can be used to update the inverse of a spectral matrix for adaptively tracking nonstationary noise fields. Such information would be required to do Bayesian location in a correlated noise field (for example, at the subarray level).

DD FORM 1473

1 NOV 65

UNCLASSIFIED

Security Classification



## KEY WORDS

## LINK A

## LINK B

## LINK C

ROLE

WT

ROLE

WT

ROLE

WT

Large-Aperature Seismic Array  
Large-Array Signal and Noise Analysis  
Iterative Techniques  
Frequency-Domain Filter Sets  
Gradient Methods  
Adaptive Update Methods  
Steepest-Descent Method  
Exact Inverse Method  
Nonstationary Noise Fields